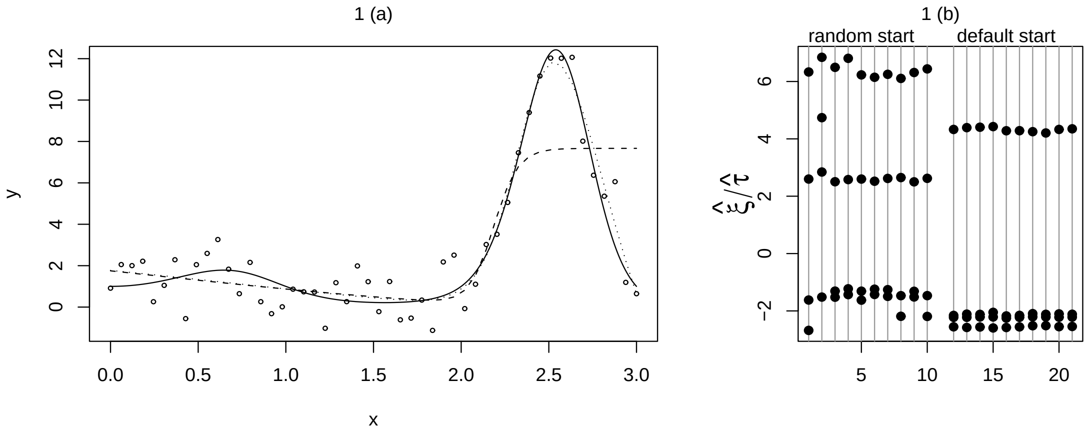

Figure 1: A simulated example. Figure 1(a) gives a simulated realization of 50 training cases, the true function $f(x)$ (—) and two local optima $\hat{f}(x)$ ($\cdots$, $- - -$) identified by different versions of ALB with $K = 4$. Figure 1(b) plots estimated reference points $\hat{\xi}_1, \ldots, \hat{\xi}_4$ (standardized by $\hat{\tau}$) from 20 runs of the algorithm, using either random starts or the default algorithm.

Hugh A. CHIPMAN and Hong GU

1. INTRODUCTION

We congratulate the author on an interesting, broad, and practical approach to flexible regression. The paper includes many features one would expect from a methodology which has been around much longer. The ability to apply the method to large datasets is appealing, the standard errors a useful addition, the ability to do quantile and/or robust regression quite convenient, and there are many extra options, such as the ability to reduce dimensionality of the predictor space. We were struck by how many avenues for further development were either already developed or suggested in the paper. The paper also raises many interesting questions and should provide fertile ground for further research.

In this discussion we consider two modifications of the algorithm. In Section 2 we look at how to deal with local optima of the parameters, and in Section 3 we modify the ALB algorithm to fit radial basis functions. Section 4 concludes with an assortment of other comments.

2. IMPROVING THE SEARCH

Local optima can be a problem for the stochastic approximation algorithm, especially if some optima fit poorly. The example in this section suggests that increased randomization of start points and stepwise deletion of bases can be useful in finding good local optima.

Figure 1(a) gives the example used to explore these strategies, and shows two local optima of the ALB function with $K = 4$. We take $f(x) = \exp\{x \sin(\pi x)\} + \epsilon$, with $\epsilon \sim N(0, 1)$, and the training set having 50 equally spaced x values in the interval (0,3). The test set is the same 50 x values, with responses $y_i = f(x_i)$ instead of $f(x_i) + \epsilon$.

Using the default parameters of the algorithm, $K = 5$ was usually chosen, providing quite an accurate fit (R^2 for test set ≈ 0.98). Closer inspection revealed that for $K = 3, 4$ the estimated function fit poorly ($R^2 \leq 0.70$). With $K = 3$, it is possible to represent a single bump, such as the large one near $x = 2.5$. However, the algorithm tended to get stuck in poor local optima ($- - -$

in Figure 1(a)), perhaps due to minimal variation in the ten sets of starting values chosen by the vector quantization (VQ).

We considered random starting points for the parameters, with the hope that increasing variability will allow the algorithm to avoid poor local optima. We set $\gamma_m = 0$, and drew random (x_i, y_i) pairs from the training set $(i = 1, \ldots, 4)$. We set $\delta_i = y_i$ and $\xi_i = 2x_i$. The x_i values were doubled because in the default runs of the algorithm, the ξ values were often outside the range of x. Using one simulated dataset, the optimization algorithm was run 100 times with different random number seeds and $K = 4$. All 100 $\hat{f}$ curves correctly identified the bump at $x = 2.5$ ($\cdots$ in Figure 1(a)). Without randomization, the original algorithm missed the bump 96 times ($- - -$) and found it the other four ($\cdots$).

Although our implementation is primitive, an increase in the randomization of initial parameter values seems to help the algorithm find better local optima. This better optimum is also found by about half the runs of the original algorithm if the initial step size in the vector quantization algorithm is doubled. A larger step size can be thought of as increasing the randomness of the algorithm, since the VQ algorithm samples the training cases one at a time in random order. Other randomization strategies, such as running VQ on small samples from the training data, might also prove successful. Simulated annealing might also be useful, although this would mean a substantial modification to the code.

In this example, we also found stepwise deletion of basis functions helpful. In Section 4.2, the author comments that because the parameters of the model must be simultaneously optimized, stepwise addition or deletion of basis functions is not used. We think that in some cases, such as when several ξ are very close, that deletion may be helpful. Consider the ξ values in Figure 2, generated by 10 runs of the default algorithm and 10 runs with random starts. We standardize values by $\hat{\tau}$ which differs for each run. The model identified by the default algorithm fits poorly, and has a group of three ξ around -2. In the reference point formulation (4) of $\phi_k(\mathbf{x})$, if two reference points are equal (say $\xi_1 = \xi_2$), then one basis function is redundant, since $\phi_2(\mathbf{x}) = \phi_1(\mathbf{x}) \exp(\gamma_2 - \gamma_1) = c\phi_1(\mathbf{x})$.

The near-duplication of reference points suggests a stepwise deletion strategy: If a model with reasonable fit has reference points that are quite close, delete one of the "near-duplicate" bases, and use the remaining parameters as starting points for the algorithm. For the current example, one run of the default algorithm with $K = 5$ bases produced reference points $\xi = -1.183, -.473, -.472, 1.524, 2.646$. By deleting $\xi_2 = -.473$ and setting $K = 4$, the default algorithm identified a solution similar to the $\cdots$ curve in Figure 1 (a). This solution offered comparable fit to the $K = 5$ case.

This illustration of two strategies for finding better optima should not be taken as an indication that the default algorithm fails - after all, $K = 5$ basis functions with good fit are identified. It does indicate however, that the search for good parameter values can still be refined in some situations, perhaps leading to more parsimonious models.

3. RADIAL BASIS FUNCTIONS

The flexibility of the ALB family of models leads naturally to comparisons with other flexible models, such as radial basis functions or neural networks. In this section we modify the stochastic approximation algorithm to estimate a radial basis function (RBF) model (Moody and Darken 1989). We consider the following parameterization of radial basis functions, as mentioned in the paper:

$$\phi_k(x) = \exp(-\tau_k^{-2}||x - \xi_k||^2) / \sum_{m=1}^{K} \exp(-\tau_m^{-2}||x - \xi_m||^2). \tag{1}$$

The parameter γ_k from ALB is dropped, and τ is allowed to vary across basis functions. As in ALB, a normalizing denominator is used. By allowing the radius τ_j of the jth basis to vary, the curvature of the function can be adjusted. Now we have

$$\partial\phi_m/\partial\tau_k = \begin{cases} 2\tau_k^{-3}||x-\xi_k||^2\phi_k(1-\phi_k) & \text{if } m=k \\ -2\tau_k^{-3}||x-\xi_k||^2\phi_k\phi_m & \text{if } m\neq k \end{cases}$$

As with γ_k in ALB, increasing τ_k increases the influence of ϕ_k relative to other basis functions. Using the same gain a_m as defined in the paper, the updating formulae in (10) become:

$$\begin{aligned} \delta_k: &\quad \delta_k + a_m^\delta h_k(x,y,\theta), \\ \tau_k: &\quad \tau_k + a_m^\gamma h_k(x,y,\theta)\delta_k - f_K(x)||x-\xi_k||^2(2\tau_k^{-3}), \\ \xi_k: &\quad \xi_k + a_m^\xi h_k(x,y,\theta)\delta_k - f_K(x)(x-\xi_k)(2\tau_k^{-2}). \end{aligned} \tag{2}$$

Note that a_m^γ is used as the gain for τ. We used these updating formulae to modify the fortran code provided by the author to estimate radial basis functions.

As mentioned in Section 2, if $\xi_k = \xi_m$ for some $k \neq m$, one basis becomes redundant. This redundancy does not occur with radial basis functions, since if $\xi_k = \xi_m$ but $\tau_k \neq \tau_m$ a mixture of two bases with different radii results.

The accuracy of RBF and ALB was compared in simulation experiments for the following functions:

1. $2\exp\{-(x_1^2+x_2^2)/2\} + 3\exp\{-(x_1^2+x_2^2)/5\}$.
2. $2\exp\{-(x_1^2+x_2^2)/2\} + 3\exp\{-[(x_1-1)^2+(x_2-1)^2]/5\}$.
3. ALB: $(2f_1+3f_2)/(f_1+f_2)$,
   where $f_1 = \exp\{1-(x_1^2+x_2^2)/4\}$ and $f_2 = \exp\{2-[(x_1-1)^2+(x_2-1)^2]/4\}$.
4. RBF: $(2f_1+3f_2)/(f_1+f_2)$,
   where $f_1 = \exp\{-(x_1^2+x_2^2)\}$ and $f_2 = \exp\{-[(x_1-1)^2+(x_2-1)^2]/4\}$.

5.–8. Examples 2, 3, 6, 7 from the paper

For each example, ten realizations of the dataset are simulated, and ALB and RBF models fit to each dataset. Table 1 gives average K and IPSE values, and also the results of paired t tests to compare IPSE values of the two models. A negative t statistic indicates that RBF has better accuracy (lower IPSE).

For examples 1, 2 and 4, RBF significantly outperforms ALB, which one would expect when the true function is of the RBF form. For the ALB function in example 3, ALB did slightly better than RBF. In example 8, there is no significant difference. In other examples, ALB outperformed RBF. Does this mean ALB should be chosen over RBF? Not necessarily. In modifying the ALB algorithm to estimate a RBF model, we changed only the updating formulae and the basis functions. Other components of the ALB algorithm, which have been carefully optimized for the ALB function (eg the gains functions a_m), were left unchanged. The performance attained by RBF using a relatively straightforward modification of the algorithm is promising, and indicates the effectiveness of the stochastic approximation algorithm.

4. OTHER COMMENTS

The ALB model is affine invariant, in the sense that if any affine transformation is applied to the predictors, there exists an ALB model using the transformed variables that provides exactly the same predictions as an ALB model using the original variables. This doesn't necessarily mean that the estimation algorithm can find this equivalent model, especially since there can be many local optima. A related issue is the fact that the algorithm is based on the reference point parameterization of the basis functions. By using Euclidean distance from reference points ξ_k in

Table 1: Comparisons between RBF and ALB: average and standard deviations from 10 replicated samples of size n

				ALB		RBF			
No.	d	n	σ_f/σ_ϵ	$\hat{K}$	IPSE	$\hat{K}$	IPSE	t	p-value
1	2	100	1	4.2	0.59(.03)	2.1	0.54(.03)	-4.08	0.003
	2	100	2	4.9	0.24(.02)	2	0.21(.008)	-4.16	0.002
	2	100	3	5	0.12(.005)	2.2	0.11(.003)	-10.15	0.000
2	2	100	1	3.7	0.59(.02)	2	0.54(.02)	-5.3	0.000
	2	100	2	4.9	0.24(.01)	2.5	0.23(.008)	-2.06	0.069
	2	100	3	5	0.12(.006)	3.1	0.11(0.006)	-3.01	0.015
3	2	100	1	2	0.52(.02)	2.1	0.53(.02)	1.84	0.098
	2	100	2	2	0.21(.007)	2.2	0.21(.009)	2.30	0.047
	2	100	3	2	0.10(.003)	2.2	0.11(.005)	2.27	0.050
4	2	100	1	4.1	0.59(.03)	2.2	0.53(.02)	-6.76	0.000
	2	100	2	5	0.24(.009)	2.2	0.21(.008)	-9.01	0.000
	2	100	3	5.2	0.12(.005)	2.1	0.11(.005)	-8.00	0.000
5	2	100	0.9	4.6	0.68(.02)	4.1	0.73(.04)	3.44	0.007
	2	200	0.9	5.8	0.65(.03)	4.5	0.69(.04)	4.53	0.001
	2	400	0.9	6.5	0.60(.02)	6.2	0.64(.02)	6.04	0.000
6	2	100	1.9	5.5	0.33(.03)	5.1	0.37(.01)	3.38	0.008
	2	200	1.9	8	0.28(.03)	7.6	0.32(.02)	6.89	0.000
	2	400	1.9	9.2	0.24(.008)	12.1	0.27(.01)	5.73	0.000
7	5	50	4.9	4.7	0.16(.04)	4.5	0.23(.05)	3.03	0.014
	5	100	4.9	5.2	0.08(.01)	5.7	0.10(.015)	6.14	0.000
	5	200	4.9	5.9	0.06(.005)	6	0.08(.007)	6.68	0.000
	10	50	4.9	2.7	0.36(.08)	3	0.38(.09)	1.05	0.32
	10	100	4.9	5.1	0.18(.07)	4.9	0.23(.07)	2.56	0.03
	10	200	4.9	5.7	0.09(.01)	6.3	0.11(.03)	3.45	0.007
8	4	25	3	2.3	0.29(.10)	2.5	0.31(.14)	0.34	0.74
	4	50	3	2.9	0.17(.04)	2.5	0.16(.02)	-0.65	0.52
	4	100	3	3.3	0.12(.009)	3	0.13(.012)	1.17	0.27
	4	200	3	3.5	0.11(.004)	3.8	0.12(.007)	2.72	0.023

the covariate space, the accuracy of the fit is sensitive to multicollinear covariates. Under the affine transformation $z = B^{'}x$, where B is invertible, the fits of the ALB regression based on the Euclidean distances in the x-space and in the z-space generally won't have the same accuracy. Elements of the algorithm such as the update steps may be affected, potentially yielding different models, even though the two forms of basis functions are one-to-one correspondent. Any sensitivity that ALB has to affine transformations should be smaller than for methods that assume additivity, such as MARS.

The inclusion of standard errors in Section 5 is a nice addition to the paper, allowing inference about the shape of the surface. The standard errors are obtained conditional on the number of bases (K), when in fact K is estimated from the data. Accounting for uncertainty in K might be accomplished via the bootstrap or a more complex Bayesian approach (such as Smith and Kohn 1996, or Chipman, George and McCulloch 1998). Bayesian (eg Draper 1995) or Bootstrap (Breiman 1996) model averaging might also improve predictions by combining multiple models. It's difficult to say whether model averaging will offer much of a gain with this form of model. Improvements are usually largest for families of models that are sensitive to small changes in the data, such as trees.

The stochastic approximation algorithm has been constructed so that the number of steps of the algorithm does not depend on the sample size. With sample sizes of more than a few hundred thousand, many points will never be used. This has a similar flavour to training the model on a sample of the data, a common technique for large datasets.

In Section 2.5, the paper uses principal components of the gradient sum-of-products matrix **G**, suggesting that if the first two eigenvalues are large, a two dimensional plot will represent most of the variation in the response model. We wonder whether this strategy could be taken further, using the directions defined by the eigenvectors to reduce the dimensionality of the original problem, perhaps yielding better models. This might also be an effective means to accomplish variable selection, eliminating variables with loadings near zero in all large principal components.

REFERENCES

Breiman, L (1996), “Bagging Predictors”, *Machine Learning*, 26, 123–140.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998) “Bayesian CART Model Search (with discussion)”, *Journal of the American Statistical Association*, 93, 935–960.

Draper, D. (1995) “Assessment and Propagation of Model Uncertainty”. *Journal of the Royal Statistical Society, Series B*, 57, 45–97.

Smith, M. and Kohn, R. (1996) “Nonparametric Regression using Bayesian Variable Selection”, Journal of Econometrics, 75, 317–344.